Privacy-enabled Remote Health Monitoring Applications for Resource Constrained Wearable Devices

Davy Preuveneers and Wouter Joosen
iMinds-DistriNet-KU Leuven
Leuven, Belgium
firstname.lastname@cs.kuleuven.be

ABSTRACT

Recent computing paradigms like cloud computing and big data have become very appealing to outsource computation and storage, making it easier to realize personalized and patient centric healthcare through real-time analytics on user data. Although these technologies can significantly complement resource constrained mobile and wearable devices to store and process personal health information, privacy concerns are keeping patients from reaping the full benefits. In this paper, we present and evaluate a practical smartwatch based lifelog application for diabetics that leverages the cloud and homomorphic encryption for caregivers to analyze blood glucose, insulin values, and other parameters in a privacy friendly manner to ensure confidentiality such that even a curious cloud service provider remains oblivious of sensitive health data.

CCS Concepts

•Security and privacy → Access control; Privacy protections; •Applied computing → Health care information systems; •Human-centered computing → *Mobile devices;*

Keywords

Health, security, privacy, access control

1. INTRODUCTION

With a growing population of patients, the pressure on health and welfare systems will continue to increase. The ever-growing healthcare costs are sparking an interest with policymakers, academics, and providers of e-health solutions to improve the efficiency of healthcare service delivery and to transform the way we go about personalized healthcare. Significant strides in wireless medical and environmental sensors promise to deliver patients and healthcare professionals with novel cost-effective solutions to health management anytime and anywhere, truly enabling the vision of mobile and pervasive healthcare.

Internet technology and mobile applications have enabled individuals to create and share increasingly complete digital traces of their daily lives. The smartphone can already provide us with an extremely granular snapshot of our current health, by monitoring our heart rate, diet and level of exercise, etc. Despite the many advances in sensor technology and telemonitoring, adoption is slow. This observation is not only triggered by the inherent cost and complexity of engineering such systems, but also because of security and privacy concerns [11, 7]. Indeed, the consolidation of data through the adoption of electronic medical record (EMR) and personal health record (PHR) systems, and the increasing need for exchanging information between patients and healthcare providers and caregivers amplifies the need for greater information *security and privacy*. This is the main challenge we address in this research.

In this work, we present a case study of a federated mobile cloud healthcare application for diabetes patients that puts them in control of their health information, and allows them to share information in a privacy-aware manner. This proof-of-concept mobile application is partly deployed on a resource constrained smartwatch, and interacts with a third party cloud service for data *storage and processing* in a federated configuration conceptually similar to Microsoft's HealthVault[1] online service. Our mobile cloud application also collects and centralizes data that would otherwise reside in disparate systems, and relies on user and device *authentication* to mitigate impersonation attacks. The contribution of this work is a practical realization of remote health data analysis application with the additional advantage of preventing eavesdropping by the health service provider in the cloud, i.e. where this collecting party remains oblivious as to what sensitive personal health information has been transferred, stored or processed. It achieves this objective by leveraging *homomorphic encryption* building blocks. While several works [12, 2, 9] discuss the theoretical opportunities of applying fully homomorphic encryption (FHE) in the e-health cloud, this applied research is one of the few attempts discussing a concrete and practical implementation of a mobile and wearable proof-of-concept involving 2 realistic application scenarios with multiple stakeholders.

After reviewing related work in section 2, we present in section 3 our context-aware diabetes monitoring assistant as a motivating example for a mobile wearable e-health application. Section 4 discusses the *Privacy by Design* princi-

[1]https://www.healthvault.com

SAC 2016, April 04-08, 2016, Pisa, Italy

DOI: http://dx.doi.org/10.1145/2851613.2851683

ples that were adopted during the proof-of-concept implementation of this application which is outlined in section 5. In section 6 we evaluate the practical feasibility of the approach and highlight some practical experiences with the development and the usage of the application. We conclude in section 7 summarizing the main insights and identifying possible topics for future work.

2. RELATED WORK

Enabling security and privacy for e-health systems is a research challenge that received wide attention the past decade, both from a technical and end-user perspective. Rodrigues et al. [7] and Wilkowska et al. [17] have elicited and analyzed numerous security and privacy requirements for cloud-based solutions of electronic health record systems. These lists of requirements have been a driving motivation for our work.

In [15], Riedl et al. present PIPE (Pseudonymization of Information for Privacy in e-Health), a health record system with the objective to address privacy through the use of pseudonyms, without having to rely on centralized patient pseudonyms lists, life-long pseudonyms or the concealment of algorithms. The PIPE security hull architecture conceals patient data through encryption, surrounded by an authentication layer in the outer hull, and an access control layer with user permissions as an inner hull. From a cloud perspective, PIPE is suitable for information storage and retrieval scenarios, but cannot handle cloud computing use cases where computation and data analysis is outsourced to an untrusted third party cloud service provider.

Löhr et al. [11] present a secure e-health comprehensive infrastructure based on trusted virtual domains (TVD) to ensure fundamental security and privacy properties, and trusted hardware components on the client. Their infrastructure leverages logically isolated execution environments, trust relationships, policy enforcement, secure communication and storage, attestation, etc. Privacy-aware processing medical data analytics is outside the scope of this work, and as with the previous work, the authors do not provide any performance results of a real deployment to assess the overall practical feasibility of their solution.

Bos et al. [2] investigate private predictive analysis on encrypted medical data. Their fully homomorphic encryption use cases focus on computing prediction functions for cardiovascular diseases and equations to screen for diabetes. These prediction functions can be expressed as truncated Taylor series on encrypted numbers, and can therefore be computed with additions and multiplications.

In [9, 8, 12], Kocabas and colleagues also assess the use of homomorphic encryption for health monitoring in the cloud. Their health use case aims to compute a.o. the average heart rate in the cloud in near real-time. For the fully homomorphic encryption (FHE), the authors leverage the HElib implementation [6]. As with the previous work, computing the average value can be carried out with addition and multiplication operators built into HElib.

Compared to the above HElib-based works, our proof-of-concept discussed in section 3 supports comparisons with caregiver defined thresholds, making our use case more useful but also more sophisticated to realize. Furthermore, our application also uses a thread-safe variant of HElib to use multiple CPU cores of the cloud server. We will evaluate the impact on the performance of these features.

Figure 1: The mobile diabetes application running on the Omate TrueSmart watch and glucometer

3. DIABETES AS A MOTIVATING CASE

In previous work [14], we presented a mobile healthcare case study of people diagnosed with type 2 diabetes using a smartphone for mobile health management. The specific aim of this application was to capture the relevant user context and annotate the health data logs in order to improve similarity measurements with previous health situations in order to help find trends and advise the individual more accurately and tailored to his current situation. In this work, we focus on the use of wearable devices for health monitoring in combination with cloud services to process and share data in a secure and privacy-sensitive way with physicians or other caregivers (e.g. parents of diabetic children). As illustrated in Figure 2, the objective is store data encrypted in the cloud such that caregivers can analyze the data without having to first download, then decrypt, and finally process all the data on their own device or workstation.

Compared to the smartphone ecosystem that we targeted in our previous work, wearable devices like smartwatches have several benefits, but they also have unmistakable drawbacks.

- Smartwatches are physically closer to the user − also at night − allowing for better activity recognition.
- Several wearables (e.g. the Motorola Moto 360 and the Apple Watch) are equipped with dedicated sensors to monitor the heart rate.
- Wearable devices have limited storage and processing capabilities, and a fairly low battery life time.

In Figure 1 we illustrate our revised mobile application running on the Omate TrueSmart Android smartwatch. As most fitness trackers, smartwatches usually include a three-axis accelerometer that we use to measure body movements, count steps, and calculate the amount of calories burnt over the course of a day. Our wearable device also features a magnetometer, a GPS, a vibrator, a microphone and an audio speaker. Furthermore, this device does not rely on a smartphone that acts as a gateway to the internet. Instead, the

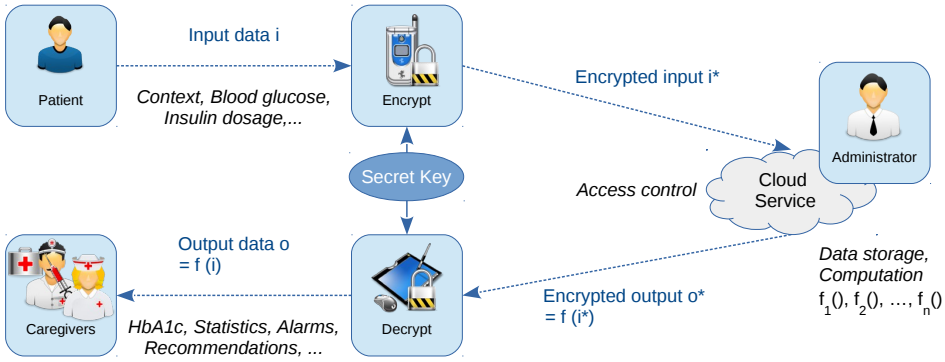

Figure 2: Privacy-aware mobile cloud computing with fully homomorphic encryption

device has its own 3G, WiFi and Bluetooth connectivity capabilities to communicate with other peripherals and cloud services. Figure 1 also shows the OneTouch UltraEasy glucometer, which we equipped with a Bluetooth extension to communicate wirelessly with the Omate TrueSmart watch.

Beyond the fingerstick glucometers, we also envision scenarios where continuous glucose monitors (CGM) are connected to a Bluetooth LE enabled smartwatch. Such CGMs are sometimes linked to insulin pumps and mobile devices in an effort to build an *artificial pancreas* [4].

4. PRIVACY REQUIREMENTS

The objective of our work is to address security and privacy challenges with *storing* and *processing* electronic health records (EHR) in the cloud [13]. This effort is in line with regulatory requirements, such as the ePHI (electronic Protected Health Information) policies of the HIPAA (Health Information Portability and Accountability Act), to only grant access to authorized users with need-to-know privileges and to make it illegal to store EHRs off-site in an unencrypted way. However, simply encrypting health records before they are sent to the cloud service nullifies the benefits of processing data in the cloud, as the cloud service provider should never have the decryption key. Our aim is to also guarantee confidentiality w.r.t. this stakeholder such that any administrator remains oblivious to the sensitive data it stores and processes, as depicted in Figure 2.

As our diabetes self-management application evolved from a standalone mobile application towards a federated mobile cloud application, we adopted *Privacy by Design* [10, 3] principles to reengineer our solution by embedding privacy in its design. The mobile cloud application for diabetes patients enforces end-to-end information security and privacy in a proactive way, allowing caregivers to process data in the cloud while respecting the patient's confidentiality needs with regards to the cloud service provider, hence creating a win-win situation for all the stakeholders involved.

In the remainder of this section, we will use 2 brief application scenarios with multiple stakeholders that serve as examples to illustrate how data is processed in the cloud in a privacy sensitive manner. The assumption is that the patient uploads blood glucose and insulin dosage values and other context properties such as timestamped locations, calorific intake, exercise levels, etc. on a regular basis. The mobile application depicted in Figure 1 provides that functionality by embedding a.o. nutrition and Metabolic Equivalents (METS) translation tables to quantify carbohydrates consumption and energy expenditure.

- **Scenario 1**: A close relative or neighbor is notified when the patient has hypoglycemia ($<$ 70 mg/dl) or hyperglycemia ($>$ 240 mg/dl), or when all the blood glucose values before a meal are outside the target range of 80-150 mg/dl for the 4 previous blood glucose measurements.

- **Scenario 2**: The physician is informed when the glycated hemoglobin (HbA1c) value goes above 7%. This value is derived from the estimated average glucose (eAG) in mg/dl over a period of 8-12 weeks:

$$eAG = 28.7 \times A1C - 46.7 \quad (1)$$

$$Average\ Whole\ BG = eAG/1.12 \quad (2)$$

  The above formulas are based on plasma glucose tests from the international A1C-Derived Average Glucose Glucose (ADAG) trial involving 507 adults [1], and the fact that whole blood glucose tests are approximately 12% lower than plasma glucose tests [16].

The above scenarios are kept simple on purpose to illustrate how encrypted data can be stored and processed in the cloud, to evaluate the practical feasibility of the approach, and to compare against alternative solutions.

5. PROTOTYPE IMPLEMENTATION

This section discusses how the health analysis is carried out on encrypted health data in the cloud.

5.1 Encrypted computations in the cloud

In our mobile cloud application, we use fully homomorphic encryption [5] so that each bit of a health parameter − blood glucose, insulin, carbs, physical activity, etc. − is encrypted. This encryption takes place on the wearable device, and the corresponding ciphertext is then sent to the cloud service for storage and processing (see Figure 2).

We will illustrate how fully homomorphic encryption works using a simple but slow symmetric scheme. Such a scheme would encrypt each bit $m \in \{0, 1\}$ individually using a large odd numbered secret key p into a ciphertext c:

$$c = \text{Encr}(m) = pq + 2r + m \quad (3)$$

Note that if p is not odd numbered, then the plaintext bit m can be reconstructed simply by computing m as $c\ mod\ 2$.

The integers q and r are randomly chosen, with the noise r chosen such that $|2r| < p/2$. The random noise r guarantees that repeated encryption of the same bit m will result in different ciphertexts c, such that after encryption similar health parameters cannot be linked.

The ciphertext c can then be decrypted back into the original plaintext bit m if one knows the secret key p:

$$m = \text{Decr}(c) = (c \bmod p) \bmod 2 \tag{4}$$

We leave it up to the reader to verify that this simple fully homomorphic encryption scheme supports addition and multiplication on the ciphertext:

$$m' + m'' = \text{Decr}(\ \text{Encr}(m') + \text{Encr}(m'')\ ) \tag{5}$$

$$m' * m'' = \text{Decr}(\ \text{Encr}(m') * \text{Encr}(m'')\ ) \tag{6}$$

To end with a more practical solution, we would need to use multiple bits to encode various health parameters (e.g. 10 bits to encode the blood glucose values, 6 bits for the insulin dosage, etc.). Additionally, our solution uses an asymmetric encryption scheme, where all the stakeholders − including the cloud service provider − have access to the public key, but not necessarily to the private key. Using an asymmetric encryption scheme, each party can encrypt numbers with the public key, but only authorized users can decrypt the ciphertext with the private key (either the original data or the outcome of a computation on the encrypted data).

5.2 Comparing encrypted values

Note that the cloud service provider cannot distinguish ciphertext messages c that are the result of encrypted bit values of 0 and 1. We use the following equations and computations modulo 2 to compare different bit values:

$$a = b \quad \Leftrightarrow \quad a + b = 0 \quad \Leftrightarrow \quad a + b + 1 = 1 \tag{7}$$

$$a > b \quad \Leftrightarrow \quad ab + a = 1 \tag{8}$$

The same equations also hold true in the encrypted domain. For example, to test whether $a > b$ we compute f as follows:

$$f = \text{Decr}(\ \text{Encr}(a) * \text{Encr}(b) + \text{Encr}(a)\ ) \tag{9}$$

After verifying whether $f = 1$, we know whether the comparison $a > b$ was true or false. We rely on the same equations to compare multi-bit values of equal length. For example, we encode the blood glucose threshold $t = 70$ mg/dl in 10 bits as follows: 00 0100 0110. To compare with a self-monitored glucose value $i = 63$ mg/dl (or in binary format 00 0011 1111), we iteratively compare the bits from the most significant one to the least significant one using equations (7) and (8). The comparison $i < t$ is true if the most significant bit of i is smaller than that of t (i.e. $i_9 < t_9$), or when these bits are equal (i.e. $i_9 = t_9$), the less significant bits are smaller. We can compare the bits pairwise from the most significant bits i_1 and t_1 down to the least significant bits i_{10} and t_{10} with the following computation:

$$\begin{aligned} o = \quad & (t_9 i_9 + t_9) + (t_9 + i_9 + 1) * [\\ & (t_8 i_8 + t_8) + (t_8 + i_8 + 1) * [\\ & (t_7 i_7 + t_7) + (t_7 + i_7 + 1) * [\\ & \quad ... \\ & (t_1 i_1 + t_1) + (t_1 + i_1 + 1) * [\\ & \quad (t_0 i_0 + t_0)\ ]]]]]]]] \end{aligned}$$

$$= 0 + 1 * [\ 0 + 1 * [\ 0 + 1 * [\ 1 + 0 * [\ ...\ ]]]]]]]] = 1 \tag{10}$$

However, the cloud service provider does not process the plaintext bits of i and t, but carries out these computations on the ciphertext counterparts $i'_{9..0} = \text{Encr}(i_{9..0})$ and $t'_{9..0} = \text{Encr}(t_{9..0})$. The caregiver has to decrypt the result $o = \text{Decr}(o^*) = 1$ using the private key to verify that indeed $i < t$. The cloud service provider cannot decrypt o^* as he does not have the private key (see Figure 2).

5.3 Implementing the scenarios

In this section, we will briefly outline how we implemented the two application scenarios based on the techniques outlined in the previous subsections.

- **Scenario 1**: Testing the 240 mg/dl hyperglycemia thresholds is carried out in a similar way as for the 70 mg/dl hypoglycemia threshold explained in the previous subsection. The mobile device of the caregiver receives f' from the cloud service provider, decrypts it into $o = \text{Decr}(o^*)$, and if $o = 1$, it alarms the caregiver.

  To check whether the 4 pre-meal blood glucose measurements were consistently outside the 80-150 mg/dl target blood glucose range, the 2 ciphertext results o^* of the $i < 80$ and $i > 150$ threshold comparisons are added up for each glucose measurement. Next, the 4 values for each meal are multiplied and the product is sent to the physician. If the decrypted value is 1, the patient had a blood glucose that was consistently out of range before each meal the previous day.

- **Scenario 2**: The cloud service provider computes the hemoglobin measure using up to 500 ciphertext blood glucose values using equations (1) and (2), and checks whether the hemoglobin value is above 7%.

  $$\begin{aligned} (1), (2) \Leftrightarrow \quad & 7 < (\text{Avg. Whole BG} * 1.12 + 46.7)/28.7 \\ \Leftrightarrow \quad & 200.9 < \text{Avg. Whole BG} * 1.12 + 46.7 \\ \Leftrightarrow \quad & 154.2 < \text{Avg. Whole BG} * 1.12 \\ \Leftrightarrow \quad & 137.7 < \text{Avg. Whole BG} \end{aligned}$$

  Our fully homomorphic encryption scheme does not handle divisions and floating point arithmetic. Rather than computing the average, it sums the encrypted blood glucose values, multiplies the 137.7 threshold with the number of measurements, encrypts the new threshold and compares the values as in scenario 1.

In the above scenarios, the cloud service cloud provider can compute the encrypted values of the blood glucose thresholds himself using the public key. If these thresholds should also remain confidential, then each caregiver has to encrypt the thresholds and send these to the cloud service provider. For the FHE algorithms themselves, our proof-of-concept mobile cloud implementation of our health application relies on the HElib [6] library.

6. EVALUATION

The objective of the evaluation is not medically nor user oriented, but rather on the performance impact and the deployment trade-offs for the mobile cloud application to assess the practical feasibility of the proposed solution.

6.1 Baseline benchmark comparison

In Table 1, we provide performance results of some baseline experiments with the HElib library [6] on three platforms. It shows the amount of time it takes to complete typical FHE operations on (1) the Omate TrueSmart watch that operates a dual-core ARM Cortex-A7 CPU running at 1 GHz, on (2) a Samsung Galaxy S4 smartphone with a quad-core ARM Cortex-A7 CPU at 1.2 GHz, and on (3) a server system with a Intel Core i7-3770 processor running at 3.40GHz. For the smartphone and smartwatch, we cross-compiled the HElib library to produce native ARM applications for Android.

- **Keypair**: Generate public and private key for $\mathbb{Z}_p$ with $p = 2$ (all computations are modulo 2).
- **Encryption**: Convert a $i = 60$ mg/dl blood glucose value and a $t = 70$ mg/dl hypoglycemia threshold into 10-bit binary representations, and sequentially encrypt them as $i^* = \text{Encr}(i)$ and $t^* = \text{Encr}(t)$.
- **Add, Multiply** and **Shift**: These are binary operators for adding and multiplying two ciphertext bits (i.e. XOR and AND respectively), and to shift a ciphertext bit vector with 1 position.
- **Comparison**: Compute $o^* = f(i^*, t^*)$ with f the comparison function $f \leftarrow a < b$ as in equation (10) using the above 3 operators.
- **Decryption**: Decrypt $o = \text{Decr}(o^*)$ and check whether $o = 1$ to conclude that indeed $i < t$.

With a ciphertext bit, we mean the encrypted representation of a single plaintext bit, not a single bit in the ciphertext.

Step	SmartWatch	SmartPhone	Server
Keypair	339800 ms	216475 ms	6990 ms
Encryption	6563 ms	3747 ms	71 ms
Add	4 ms	1 ms	0 ms
Multiply	3472 ms	1913 ms	50 ms
Shift	5912 ms	3439 ms	329 ms
Comparison	314710 ms	178926 ms	7134 ms
Decryption	2388 ms	1471 ms	138 ms

Table 1: Baseline benchmarks of a single hypoglycemia threshold comparison on 3 platforms

Table 1 shows that the comparison function is computationally expensive, even on the server. Also the sequential encryption of 2 values (t and i) on the smartwatch takes more than 6 seconds. However, the standard for glucose meters is that 95% of all test results must to be within 20% of the actual blood glucose level for results greater than 75 mg/dl, and within 15 mg/dl for values below 75 mg/dl. As shown in Table 2, by ignoring the 3 least significant bits and initialize the FHE parameters for best performance (cfr. [6] for details), we can reduce the processing time with at least a factor 2.

In the following subsection, we will discuss other optimizations to improve further the performance.

6.2 Homomorphic evaluation optimizations

The encryption, comparison and decryption steps processed individual values. However, HElib supports ciphertext packing to combine many plaintext elements in a single ciphertext and optimize homomorphic evaluation. Also, due to limitations with the library, only one CPU core was used on both mobile platforms. The server-side implementation did not use all CPU cores either. By encrypting, decrypting and comparing values in parallel on all cores, we can further optimize the performance.

Step	SmartWatch	Server
Keypair	172415 ms	11254 ms
Encryption	2914 ms	62 ms
Add	2 ms	0 ms
Multiply	1372 ms	50 ms
Shift	2626 ms	185 ms
Comparison	67633 ms	2919 ms
Decryption	880 ms	97 ms

Table 2: Results ignoring the 3 least significant bits

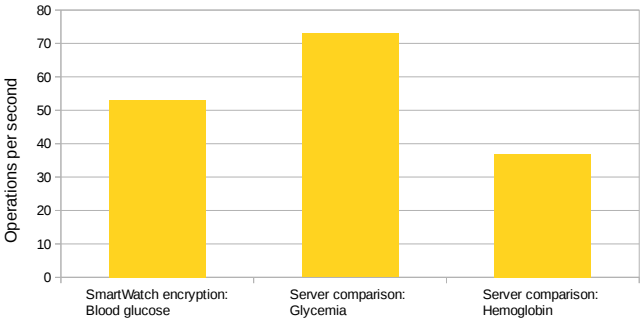

Figure 3: Parallel homomorphic scenario evaluation

For scenario 1, we evaluate all comparisons for each blood glucose result in parallel by leveraging ciphertext packing. For scenario 2, we compute the hemoglobin estimate based on 500 blood glucose measures. We parallelize the total glucose computation of all 500 measurements with the same ciphertext packing technique. Figure 3 shows the number of evaluations per second on the smartwatch and server. On the wearable, we count how many blood glucose values can be encrypted, and on the server we count the number of hypoglycemia/hyperglycemia and hemoglobin threshold computations. In this experiment, we used all the available CPU cores. These numbers are far more acceptable, as patients with fingerstick glucometers usually test their blood glucose less than 10 times per day. However, while not discussed in the paper, our proof-of-concept also processes other parameters (insulin, calorific intake, energy expenditure, etc.).

6.3 Discussion

One might argue that the mathematical operations and threshold-based comparisons are fairly simple and can be computed directly on the wearable device without leveraging the cloud. Indeed, the computations in the above use cases can be computed on the plaintext data on the mobile device, and they will most likely be less computationally intensive compared to the sophisticated encryption of the plaintext data for homomorphic computations in the cloud. However, for practical client-side data analysis the patient's device must always be online and connected so that remote third party care providers (e.g. friends, relatives) can issue requests and retrieve the health parameters. Additionally, expensive access control mechanisms would have to be put in place on the wearable device to restrict access to the sen-

sitive data. Given the resource constrained nature of these mobile and wearable devices, this is likely not feasible.

With our approach we allow for any kind of additive and multiplicative operations on the encrypted data in the cloud. Rather than embedding the computations and threshold comparisons in the mobile application, the caregiver can now define its own operations on the encrypted data without any involvement of the patient's device nor having access to all the raw sensitive data. As a trade-off, however, we need a resource rich environment like the cloud to deal with the computational complexity of our solution.

7. CONCLUSION

We presented and evaluated a practical smartwatch based lifelog application for diabetics that leverages the cloud and homomorphic encryption for caregivers to analyze health parameters in a privacy friendly manner to ensure confidentiality such that even a curious cloud service provider remains oblivious of sensitive health data. By carefully selecting the FHE initialization parameters and optimizing the parallel homomorphic evaluation, we demonstrate the practical feasibility of our solution, although there is a performance penalty of several orders of magnitude. Furthermore, computational overhead will be a concern when dealing with a large patient population using continuous glucose monitoring devices (blood glucose results every 5 minutes).

In the above scenarios, the wearable sent all data fully homomorphic encrypted to the cloud. However, for memory constrained devices this is not practical due to the resource limitations. As future work, we will investigate the performance impact of having the data on the wearable encrypted under AES. Homomorphic AES decryption would transform the AES-encrypted data into an FHE-encrypted data, which can then be used for the same computations.

Acknowledgment

This research is partially funded by the Research Fund KU Leuven and the ICON funding programme of iMinds.

8. REFERENCES

[1] Association, A. D. Standards of Medical Care in Diabetes-2014. *Diabetes Care 37*, Supplement 1 (Jan. 2014), S14–S80.

[2] Bos, J. W., Lauter, K., and Naehrig, M. Private predictive analysis on encrypted medical data. *Journal of Biomedical Informatics 50* (2014), 234–243.

[3] Cavoukian, A., Taylor, S., and Abrams, M. Privacy by design: essential for organizational accountability and strong business practices. *Identity in the Information Society 3*, 2 (2010), 405–413.

[4] Elleri, D., Dunger, D. B., and Hovorka, R. Closed-loop insulin delivery for treatment of type 1 diabetes. *BMC medicine 9* (2011), 120. Review,Research Support, Non-U.S. Gov't,Research Support, N.I.H., Extramural,.

[5] Gentry, C. *A Fully Homomorphic Encryption Scheme*. PhD thesis, Stanford, CA, USA, 2009. AAI3382729.

[6] Halevi, S., and Shoup, V. Algorithms in helib. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I* (2014), J. A. Garay and R. Gennaro, Eds., vol. 8616 of *Lecture Notes in Computer Science*, Springer, pp. 554–571.

[7] JPC Rodrigues, J., de la Torre, I., Fernández, G., and López-Coronado, M. Analysis of the security and privacy requirements of cloud-based electronic health records systems. *J Med Internet Res 15*, 8 (Aug 2013), e186.

[8] Kocabas, O., and Soyata, T. Medical data analytics in the cloud using homomorphic encryption. In *Handbook of Research on Cloud Infrastructures for Big Data Analytics*, P. R. Chelliah and G. Deka, Eds. IGI Global, Hershey, PA, USA, Mar 2014, ch. 19, pp. 471–488.

[9] Kocabas, Ö., Soyata, T., Couderc, J., Aktas, M., Xia, J., and Huang, M. C. Assessment of cloud-based health monitoring using homomorphic encryption. In *2013 IEEE 31st International Conference on Computer Design, ICCD 2013, Asheville, NC, USA, October 6-9, 2013* (2013), IEEE Computer Society, pp. 443–446.

[10] Langheinrich, M. Privacy by design - principles of privacy-aware ubiquitous systems. In *Proceedings of the 3rd International Conference on Ubiquitous Computing* (London, UK, UK, 2001), UbiComp '01, Springer-Verlag, pp. 273–291.

[11] Löhr, H., Sadeghi, A.-R., and Winandy, M. Securing the e-health cloud. In *Proceedings of the 1st ACM International Health Informatics Symposium* (New York, NY, USA, 2010), IHI '10, ACM, pp. 220–229.

[12] Page, A., Kocabas, Ö., Ames, S., Venkitasubramaniam, M., and Soyata, T. Cloud-based secure health monitoring: Optimizing fully-homomorphic encryption for streaming algorithms. In *2014 IEEE GLOBECOM Workshops, Austin, TX, USA, December 8-12, 2014* (2014), IEEE, pp. 48–52.

[13] Pearson, S. Privacy, security and trust in cloud computing. In *Privacy and Security for Cloud Computing*, S. Pearson and G. Yee, Eds., Computer Communications and Networks. Springer London, 2013, pp. 3–42.

[14] Preuveneers, D., and Berbers, Y. Mobile phones assisting with health self-care: a diabetes case study. In *Mobile HCI* (2008), G. H. ter Hofte, I. Mulder, and B. E. R. de Ruyter, Eds., ACM International Conference Proceeding Series, ACM, pp. 177–186.

[15] Riedl, B., Grascher, V., Fenz, S., and Neubauer, T. Pseudonymization for improving the privacy in e-health applications. In *Proceedings of the Proceedings of the 41st Annual Hawaii International Conference on System Sciences* (Washington, DC, USA, 2008), HICSS '08, IEEE Computer Society, pp. 255–264.

[16] Tonyushkina, K., and Nichols, J. H. Glucose meters: a review of technical challenges to obtaining accurate results. *Journal of diabetes science and technology 3*, 4 (July 2009), 971–980.

[17] Wilkowska, W., and Ziefle, M. Privacy and data security in e-health: Requirements from the user's perspective. pp. 191–201.